SPECIFICATION

Electronic Version 1.2.8

Stylesheet Version 1.0

LAYERED ARTICLE DATA VERIFICATION

Background of the Invention

[0001] Technical Field

[0002] The present invention relates generally to layered articles, and more particularly, to verification of layered article data preparation.

[0003] Related Art

[0004] There are a variety of applications where an article is used to create a product is formed by a number of design layers. Exemplary applications include: glass or metal etching and semiconductors. In these applications, data preparations are created to direct the construction of an article, e.g., a tool such as a screen(s), template(s) or mask(s), used to create the final product or for the actual final product. For instance, in terms of semiconductor masks, data preparations representative of particular treatments of design layers to be used to build the mask are created. Each application may have its own unique manner of indicating what treatments are necessary. In any application, knowing whether a data preparation will create the desired product is critical in preventing wasted resources.

[0005] Unfortunately, many of the articles used in these applications are fairly complex because of, for example, the large number of design layers, the requirement for a number of discrete segments (e.g., 10^{6}+), and different requirements for each layer and/or segment. The complexity makes test runs too time-consuming and expensive. In addition, relative to semiconductors, masks are oftentimes built in a different location than where manufacturing will occur, which further prevents test runs.

[0006] In view of the foregoing, there is a need for improved verification of a data

preparation for articles constructed of multiple design layers.

Summary of the Invention

[0007] The invention includes a system, method and program product to verify a data preparation is correct for an article constructed of a plurality of design layers. An instruction algorithm representative of the data preparation is reduced to fundamental algorithms having corresponding fundamental graphical representations. The graphical representations can be combined to form a combination graphical representation that is used to determine whether the data preparation is correct. The invention can be used to verify correct data preparation of highly complex articles.

[0008] A first aspect of the invention is directed to a method of verifying a data preparation for an article constructed of a plurality of design layers, the data preparation being stated in terms of an instruction algorithm, the method comprising the steps of: restating the instruction algorithm in terms of at least two fundamental algorithms; creating a graphical representation for each fundamental algorithm; combining the graphical representations corresponding to each fundamental algorithm according to the restated instruction algorithm to form a combined graphical representation; and determining whether the data preparation is correct based on the combined graphical representation.

[0009] A second aspect of the invention is directed to a system for verifying a data preparation for an article constructed of a plurality of design layers, the system comprising: means for restating an instruction algorithm representative of the data preparation for the article in terms of at least two fundamental algorithms; means for creating a graphical representation for each fundamental algorithm; means for combining the graphical representations corresponding to the at least two fundamental algorithms to form a combined graphical representation; and means for determining whether data preparation is correct based on the combined graphical representation.

[0010]

A third aspect of the invention is directed to a computer program product comprising a computer useable medium having computer readable program code embodied therein for verifying a data preparation for an article constructed of a

plurality of design layers, the program product comprising: program code configured to restate an instruction algorithm representative of the data preparation for the article in terms of at least two fundamental algorithms; program code configured to create a graphical representation for each fundamental algorithm; program code configured to combine the graphical representations corresponding to the at least two fundamental algorithms to form a combined graphical representation; and program code configured to determine whether the data preparation is correct based on the combined graphical representation.

[0011] The foregoing and other features and advantages of the invention will be apparent from the following more particular description of preferred embodiments of the invention.

Brief Description of the Drawings

[0012] The preferred embodiments of this invention will be described in detail, with reference to the following figures, wherein like designations denote like elements, and wherein:

[0013] FIG. 1 shows a block diagram of a verification system in accordance with the invention;

[0014] FIG. 2 shows a flow diagram of a method in accordance with the invention;

[0015] FIG. 3 shows exemplary graphical representations used by the system of FIG. 1;

[0016] FIG. 4 shows an exemplary combined graphical representation;

[0017] FIG. 5 shows a result of the combined graphical representation of FIG. 4;

[0018] FIG. 6 shows a table of correct positive preparations for a variety of fundamental algorithms; and

[0019] FIG. 7 shows a table of correct negative preparations for a variety of fundamental algorithms.

Detailed Description of the Invention

[0020] For purposes of description, the following terminology will be employed: An

"article" is something that is used to create a product; the article is created using at least two design layers. An article generally is used in an integration process or a layered manufacturing process such as semiconductor manufacture. For instance, the article may be a tool such as a mask, template, screen, etc., used to create the product. A "product" refers to an intended final object created. In terms of semiconductor manufacture, the product would be the circuitry created by the mask, or the actual wafer. As an alternative, the article may also be the product itself depending on the application. For instance, in terms of semiconductors, the article may be for the mask, while for glass etching, the article may be of the final etched-glass product. An "instruction algorithm" is a definition of the article made up of mathematical functions (e.g., group theory) of design layers. A "design layer" shall be used herein to refer to some basic shape, area or form known within a particular application that is used to create the article. Within the exemplary application of semiconductor manufacturer, a semiconductor mask may be the article and a design layer may be a generic photomask level. Alternatively, the design layer may be the layers of circuitry and the article may be the actual wafer.

[0021] With reference to the accompanying drawings, FIG. 1 is a block diagram of a verification system 10 in accordance with the invention. Verification system 10 preferably includes a memory 12, a central processing unit (CPU) 14, input/output devices (I/O) 16 and a bus 18. A database 20 may also be provided for storage of data relative to processing tasks. Memory 12 preferably includes a program product 22 that, when executed by CPU 14, comprises various functional capabilities described in further detail below. Memory 12 (and database 20) may comprise any known type of data storage system and/or transmission media, including magnetic media, optical media, random access memory (RAM), read only memory (ROM), a data object, etc. Moreover, memory 12 (and database 20) may reside at a single physical location comprising one or more types of data storage, or be distributed across a plurality of physical systems. CPU 14 may likewise comprise a single processing unit, or a plurality of processing units distributed across one or more locations. A server computer typically comprises an advanced mid-range multiprocessor-based server, such as the RS6000 from IBM, utilizing standard operating system software, which is designed to drive the operation of the particular hardware and which is compatible

with other system components and I/O controllers. I/O 16 may comprise any known type of input/output device including a network system, modem, keyboard, mouse, scanner, voice recognition system, CRT, printer, disc drives, etc. Additional components, such as cache memory, communication systems, system software, etc., may also be incorporated into system 10.

[0022] As shown in FIG. 1, program product 22 may include a restater 24, a graphical representation (rep.) creator 26, a combiner 28, a comparator 30 and other system components 32. Other system components 32 may be any complementary functions to verification system 10 not explicitly delineated below.

[0023] Operation of system 10 and implementation of the method will be described relative to FIGS. 2-7. In a first step S1, shown in FIG. 2, an instruction algorithm that represents or instructs on the construction of the desired article is restated by restater 24. As indicated in FIG. 2, this may be a reiterative process. The instruction algorithm as prepared may be referred to as a "data preparation." The restating step includes reducing the instruction algorithm to a point where it can be stated in terms of at least two fundamental algorithms. In one embodiment, the step of restating includes organizing the instruction algorithm according to group theory operators, e.g., union, intersection, exclusive. The type of operators, however, can be changed to accommodate different applications. For instance, boolean algebra may be more appropriate for some applications.

[0024] A "fundamental algorithm" is an instruction that represents a convenient restating or grouping of element(s) used in an instruction algorithm and, hence, convenient restating or grouping of design layer(s). In other words, a fundamental algorithm may represent a single basic design layer definition or a compound design layer definition that is particular to the application. For instance, in FIG. 3, fundamental algorithm 42C may represent a compound design layer definition indicated as WA+VL+VN for the particular application, i.e., C = WA+VL+VN. In addition, fundamental algorithms may include multiple elements, e.g., A = B, C - D, etc.

[0025] An exemplary instruction algorithm such as VSarticle = ((WN = VK) - VSNOT) +VSDL) may be restated using grouping fundamental algorithm A = (WN = VK) to VSarticle = ((A-VSNOT) + VSDL). VSarticle indicates an article referred to as VS and

VSDL indicates a design layer also referred to as VS. A second reiteration may use fundamental algorithm B = A - VSNOT to further restate the instruction algorithm as VSarticle = B + VSDL. A third reiteration may use fundamental algorithm C = VSDL to restate the instruction algorithm as VSarticle = B+C. A fourth reiteration may use fundamental algorithm D = B+C to restate the instruction algorithm as VSarticle = D.

[0026] In contrast, an instruction algorithm may be as simple as WNarticle = WNDL, and a restating fundamental algorithm may include A = WNDL, i.e., the fundamental algorithm is simply a restatement of a design layer. Accordingly, the instruction algorithm may be restated as WNarticle = A. Another example is shown in FIG. 3, where fundamental algorithm 42D may represent a single design layer definition indicated as VN for the particular application, i.e., D = VN.

[0027] The instruction algorithm and article can be very complex. For instance, where the article is a semiconductor mask, the instruction algorithm may include many parameters. The following table illustrates sample instruction algorithms for masks in a semiconductor manufacturer. The mask includes discrete segments 1-3, which each have their own respective instruction algorithm. Each segment could also be considered a design layer.

[t1]

Mask/Article	Segment 1	Segment 2	Segment 3
VS	WN=VK	{[(WN=VK) -VSNOT] + VS}	(WS=VK)+KERFVS +NEGMKS
PK	WN=VK	WN=VK	(WN=VK)+KERFPK +NEGMKS
XN	(XA+XP)+XN	[(XA+XP+YAK) -XN]	{[XA+XP]-XN+KERFXN-POSMKS+PROTECT+FRAME
VS	(WN=VK)	{[(WN=VSNOT]+VS}	[WN=VK]+KERFVS+ NEGMKS

[0028] In step S2, a graphical representation for each fundamental algorithm is created using creator 26. Referring to FIG. 3, each fundamental algorithm 42A-D has a corresponding graphical representation 44A-D. FIG. 4 represents a fundamental algorithm 42E that includes two elements C-D and a multiple part graphical representation 44E. Although, the graphical representations shown are rectangular

and have set cross-hatching or shades of gray coloring, it should be recognized that a variety of graphical representations can be created depending on the application. For instance, a variety of graphical representations having different shapes, shading, coloring, size, parts, etc. may be created. A convenient set of these graphical representations may be created for each application.

[0029] Actual implementation of creator 26 may take any of a variety of well-known data entry formats. For instance, creator 26 may be implemented through a graphical user interface having a graphical representation entry section in the form of a drawing system and an entry section for the corresponding fundamental algorithm.

[0030] In step S3, the graphical representations corresponding to the at least two fundamental algorithms are combined by combiner 28 according to the restated instruction algorithm to form a combined graphical representation. Combination is made by the above-mentioned operators depending on application. With group theory, combination is made using, for example, union, intersection and exclusive. For instance, referring to FIG. 4, the above-mentioned fundamental algorithms C and D may be combined according to instruction algorithm ((WA+VL+VN) - VN) to C-D. This would result in a combined graphical representation 44, as shown in FIG. 4. That is, C-D may be a fundamental algorithm or a combination of fundamental algorithms.

[0031] In step S4, a determination is made by comparator 30 whether the data preparation is correct based on the combined graphical representation. This determination may take many forms. Where the article is the product, the graphical representation may be implemented, i.e., used to create the final product electronically, and the result compared to the final product. The final product may be actually created as part of the process or by some other entity. For instance, as shown in FIG. 5, a result of implementation of the combined graphical representation of FIG. 4 is shown. Where the article is for a tool such as a mask, the combined graphical representation and desired tool may be compared to make the determination. Depending on application, the comparison can be completed visually or the desired article may be input/imported into system 10 for electronic comparison. If the combined graphical representation is correct, the user knows the instruction algorithm/data preparation is correct. Otherwise, the user knows the instruction

algorithm/data preparation requires adjustment before commitment to physical form or continuation of processing.

[0032] Depending on application, the determination step may also require additional processing by system 10. For instance, where the article is a semiconductor mask, the ultimate data preparation may need to be for a negative. Accordingly, a determination of the polarity of the article and appropriate inversion of the combined graphical representation may be required prior to the determination. This is exemplified by comparing FIGS. 6 and 7. FIG. 6 shows correct positive preparations and FIG. 7 correct negative preparations for a variety of algorithms for a mask. In FIGS. 6 and 7, 'Polarity' represents whether the article is based on a positive (CP) or negative (CN) of the product; 'CGR' represents the combined graphical representation; 'D_Prep' represents the implementation of the combined graphical representation, i.e., the mask; and 'Result' represents implementation of the D_Prep, i.e., the result on the final product. As expected, in FIG. 6, the Results are negatives of the D_Prep, while in FIG. 7, the Results are commensurate with the D-Prep.

[0033] As noted above, articles can be very complex and may include a number of discrete segments. For instance, in the above table, each segment may represent a discrete segment of a mask. Segment 3 may represent a frame feature, and segments 1 and 2 mask features to be placed within the frame feature. In this case, the above-described method may be repeated for each discrete segment within a larger article.

[0034] The above-described system and method provides a mechanism to verify data preparation for an article without wasting resources. The invention finds advantage with practically any type of application where an article is constructed of multiple design layers, e.g., etching on glass, etching on metal, mask building, etc.

[0035] Another embodiment of the invention includes a set of graphical representations as described above that can be used to verify a data preparation for creating an article having a plurality of design layers is correct.

[0036] In the previous discussion, it will be understood that the method steps discussed preferably are performed by a processor, such as CPU 14 of system 10, executing instructions of program product 22 stored in memory. It is understood that the

various devices, modules, mechanisms and systems described herein may be realized in hardware, software, or a combination of hardware and software, and may be compartmentalized other than as shown. They may be implemented by any type of computer system or other apparatus adapted for carrying out the methods described herein. A typical combination of hardware and software could be a general-purpose computer system with a computer program that, when loaded and executed, controls the computer system such that it carries out the methods described herein. Alternatively, a specific use computer, containing specialized hardware for carrying out one or more of the functional tasks of the invention could be utilized. The present invention can also be embedded in a computer program product, which comprises all the features enabling the implementation of the methods and functions described herein, and which - when loaded in a computer system - is able to carry out these methods and functions. Computer program, software program, program, program product, or software, in the present context mean any expression, in any language, code or notation, of a set of instructions intended to cause a system having an information processing capability to perform a particular function either directly or after the following: (a) conversion to another language, code or notation; and/or (b) reproduction in a different material form.

[0037] While this invention has been described in conjunction with the specific embodiments outlined above, it is evident that many alternatives, modifications and variations will be apparent to those skilled in the art. Accordingly, the embodiments of the invention as set forth above are intended to be illustrative, not limiting. Various changes may be made without departing from the spirit and scope of the invention as defined in the following claims.